PIZZA SALES ANALYSIS AND VISUALIZATION

Introduction:

The pizza industry is one of the most competitive segments in the food service sector, with rapidly evolving consumer preferences, seasonal demand fluctuations, and fierce competition. This report aims to provide a comprehensive analysis of pizza sales, identifying trends, consumer behaviors, and factors impacting revenue. Understanding these patterns allows us to make data-driven decisions that can enhance customer satisfaction, optimize inventory management, and improve profitability.

Our analysis focuses on key sales metrics such as top-performing pizza types, size preferences, customer segments, and peak sales times. Additionally, we will examine the effects of promotions, seasonal trends, and discounts (where applicable) on overall sales performance. By leveraging data insights, this report seeks to provide actionable recommendations that can help in crafting targeted marketing strategies, maximizing operational efficiency, and ultimately driving sales growth.

Objective:

The objective of this analysis is to gain insights into pizza ordering patterns and revenue contributions based on different factors such as pizza type, size, and time of day. By exploring the data and visualizing key metrics, we aim to identify the month with the highest sales, peak ordering hours, the most popular pizza types, the pizza size contributing the most to revenue, and the revenue generated by each pizza type. This analysis will provide valuable information for understanding customer preferences, optimizing inventory, and making data-driven decisions for business growth.

Import Python Libraries:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Dataset:

The data used for this analysis is sourced from a bootcamp dataset and represents pizza order information from the year 2015. This Dataset taken from kaggale website.The dataset provides detailed information on orders, order details, and pizza types. It offers a comprehensive view of customer preferences, order quantities, prices, and other relevant attributes. By utilizing this dataset, we can uncover valuable insights about pizza ordering trends and revenue contributions during the specified time period.

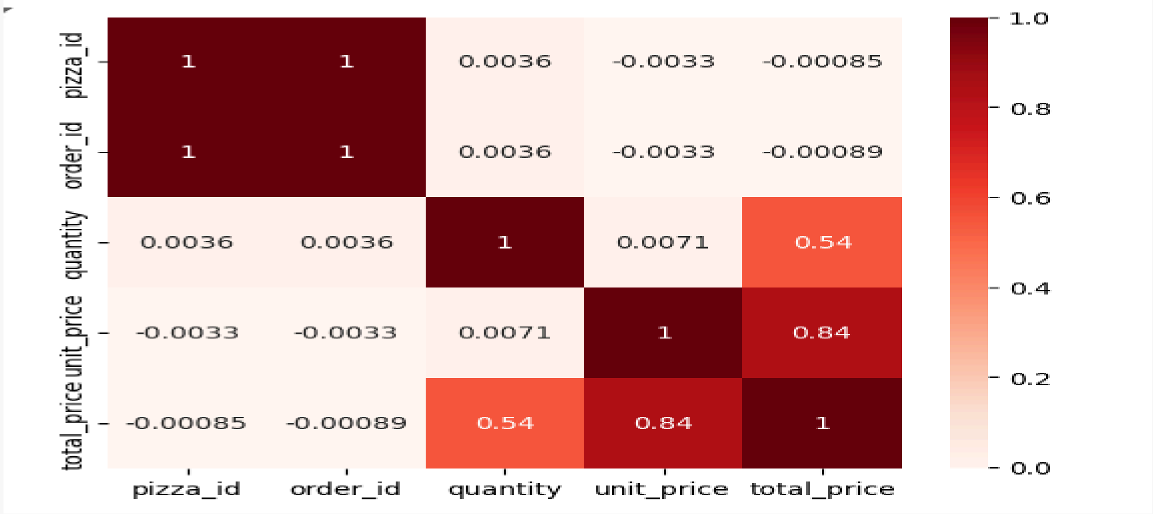
Data Cleaning and Preparation:

- **Remove Duplicates**: Ensure each transaction is unique.
- **Convert Date and Time Columns**: Standardize formats for date and time, enabling time-based analysis.
- **Handle Missing Values**: Fill missing data for relevant fields (if any).
- These initial data-cleaning steps ensured that the dataset was ready for exploration and analysis, providing a solid foundation for deriving meaningful insights.

Data Exploration and Visualization:

Heat Map:

```
# Heat Map for Correlation
numeric_columns = df.select_dtypes(include= ['number']) #number takes only
our numeric columns
sns.heatmap(numeric_columns.corr(),cmap = 'Reds', annot =True) #Red seems
right for a heatmap
plt.show()
```



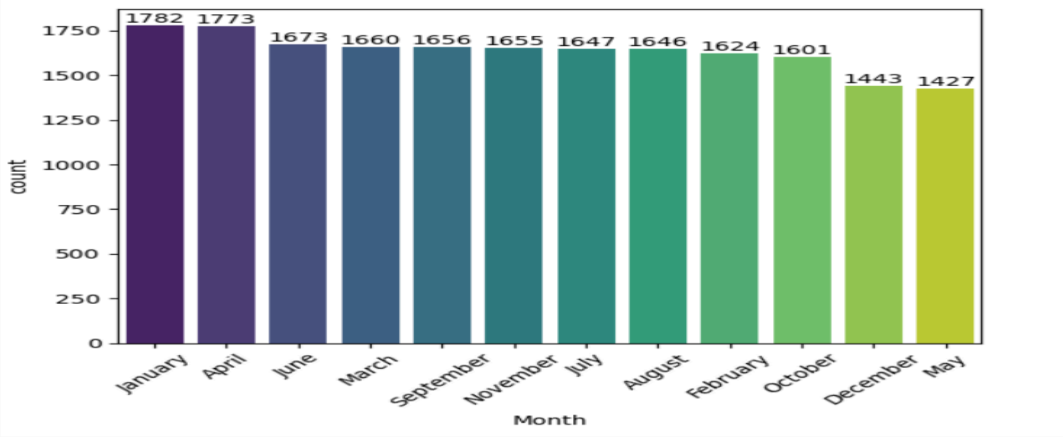
	pizza_id	order_id	quantity	unit_price	total_price
pizza_id	1	1	0.0036	-0.0033	-0.00085
order_id	1	1	0.0036	-0.0033	-0.00089
quantity	0.0036	0.0036	1	0.0071	0.54
unit_price	-0.0033	-0.0033	0.0071	1	0.84
total_price	-0.00085	-0.00089	0.54	0.84	1

To identify the month with the highest sales, we conducted data exploration using a line chart. The sales data was aggregated and plotted against the corresponding months. The line chart revealed the trend of sales over the months, allowing us to determine the month with the highest sales. The screenshot below showcases the line chart highlighting the sales trend.

```
# Add new column year
df["Year"] = df["order_date_time"].dt.year

# Add new column month
df["Month"] = df["order_date_time"].dt.month_name()

# Add new column day
df["Day"] = df["order_date_time"].dt.day_name()
```

Count Plot:

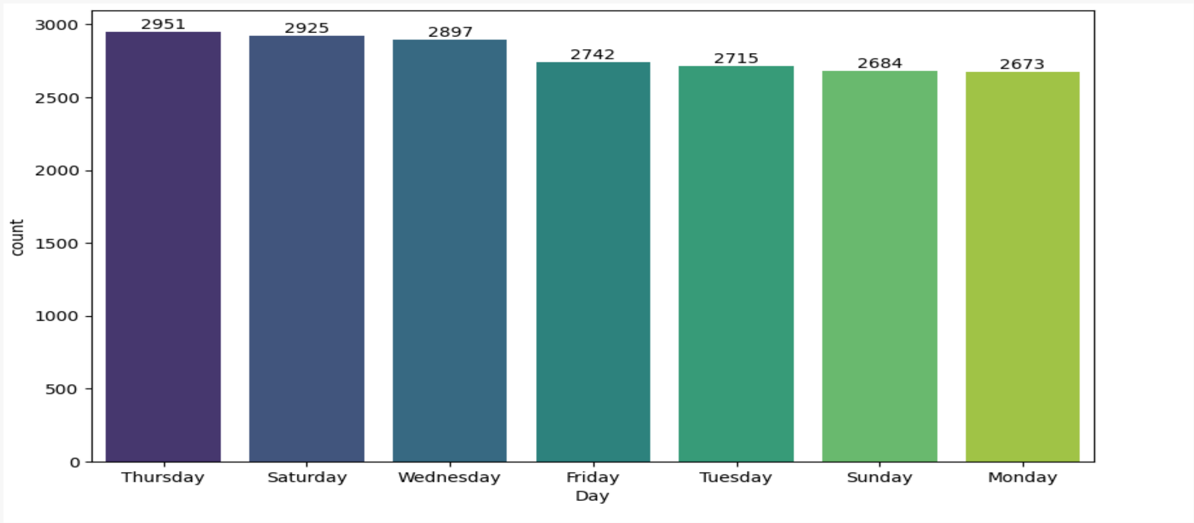
The line count_plot(x_axis="Month", rotation=45) likely creates a bar plot showing the number of items sold per month. The x_axis="Month" sets months along the x-axis, while rotation=45 rotates month labels by 45 degrees for better readability. This visualization helps quickly spot monthly sales trends.



```
# Finally we plot! Let's see the number of items sold in each month
count_plot(x_axis = "Month", rotation = 45)
```

we have some pizza loving people going out in January and April!

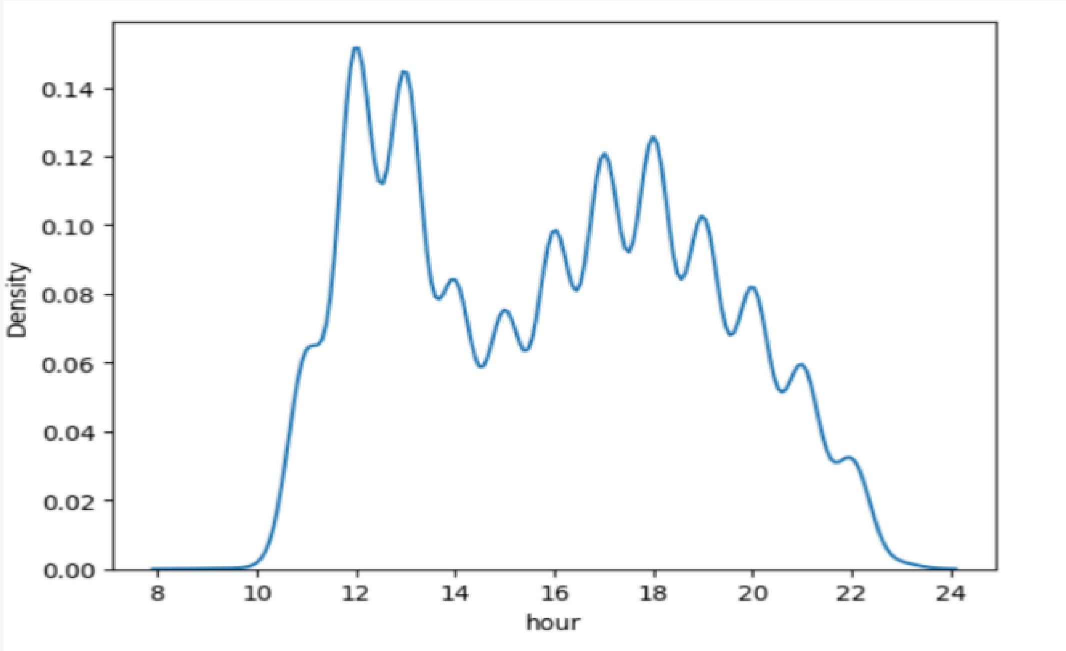
```
# Let's make the plot a littler larger for visibility
plt.figure(figsize = (10,6))

# Call our function, told you it would make life easier!
count_plot(x_axis = "Day")
```



Thursday, Saturday, and Wednesday seem to be pizza nights!

Line Plot:

```
#Let's check out the most popular hours based on sales.
#We have to change order time to a date_time value that allows for hours.
df['order_time'] = pd.to_datetime(df['order_time'], format='%H:%M:%S')

# Now extract the hour component and store it in a new column.
df['hour'] = df['order_time'].dt.hour
# see distribution of sales for hours in the day

sns.kdeplot(df["hour"])
plt.show()
```

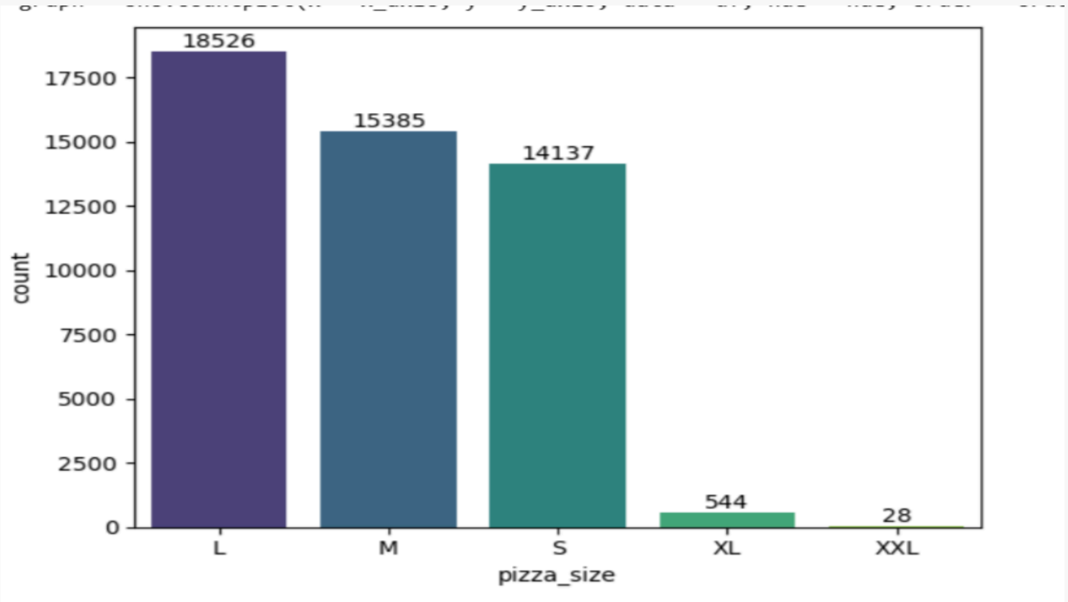


Peaking around 12 PM, 1:30 PM, and the beginning of the evening at 5:30 PM and 6:30 PM. Makes sense, that is when I would eat pizza!

Bar Plot:

In below bar Graph shown the count and pizza size based on the count we calculate the most sold pizza sizes.in this bar Graph show the Large seems to be the most sold my a significant degree. yammy

```
#We have our standard pizza sizes.
df["pizza_size"].unique()
#This one is easy, let's get some insight on the most sold pizza sizes.
count_plot(x_axis = "pizza_size")
```

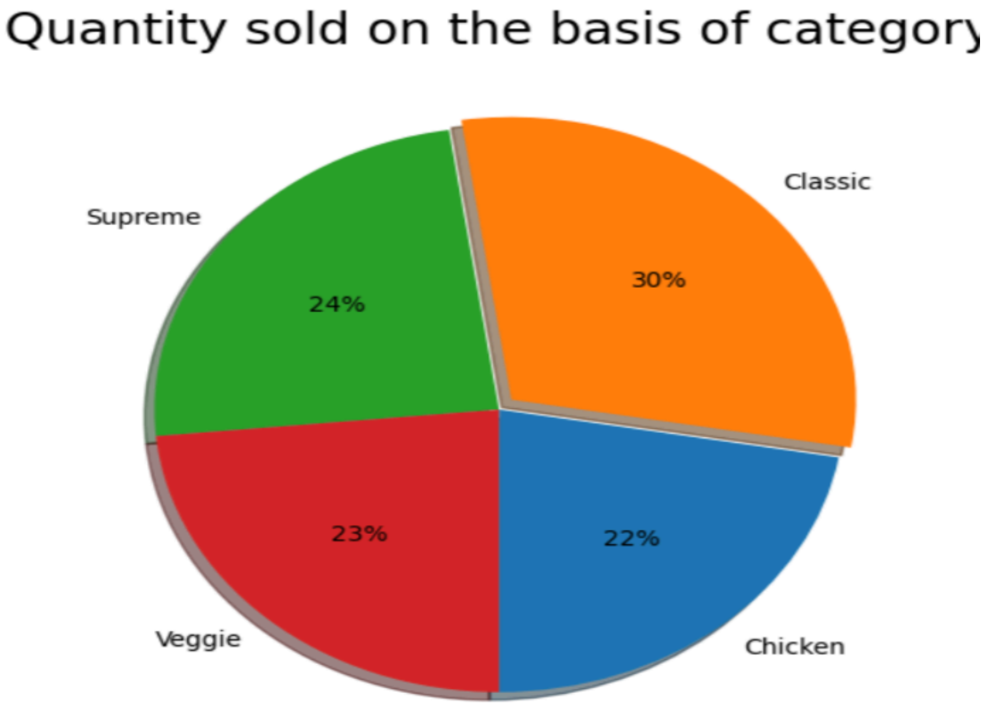


Pie Chart:

Pie chart shown the pizza category and quality. The pizza categories are Supream,classic,veggi,chicken. Based on the qulity we caluculate thequality sold on the basis of category.pie chart give result like classic has the largest market share.

```
# creation the dataframe with quantities by categories
categories = df.groupby('pizza_category')['quantity'].sum()
categories

# pieplot
plt.figure(figsize=(6, 6))
```

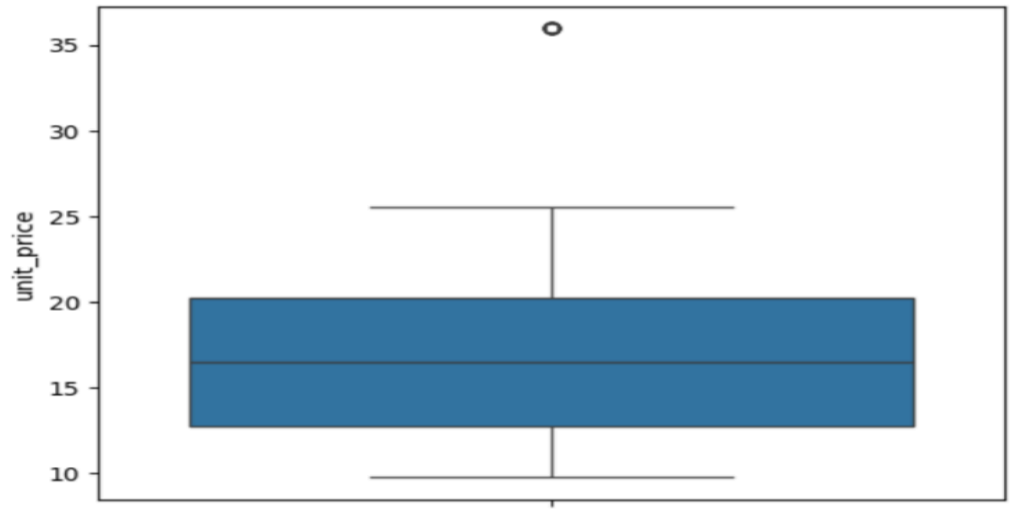
```
# color_palette = sns.color_palette("Set2")
categories = df.groupby('pizza_category')['quantity'].sum()

plt.pie(data = categories,\
        x=categories.values, labels = categories.index, \
        explode=(0,0.05,0,0), autopct='%1.0f%%', \
        shadow=True, \
        startangle=270)
plt.title("Quantity sold on the basis of category", size=20)
plt.show()
```



Box Plot:

```
#We are pulling out a new plot, the boxplot! Let's check the ranges of
pizza prices.
sns.boxplot(y = "unit_price", data = df)
```

```
<Axes: ylabel='unit_price'>
```

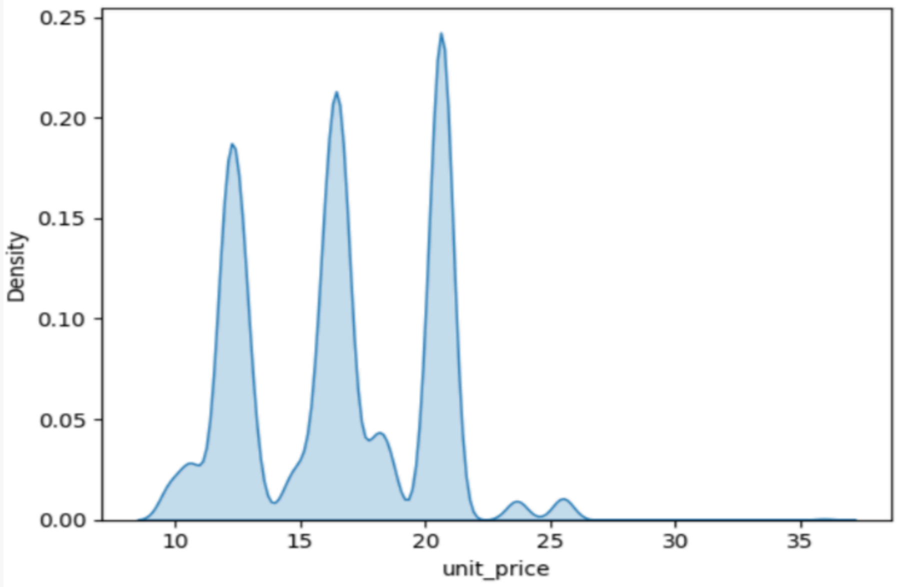


Looks like most pizza is around that 16 dollar range. We do have some big spenders though.

Density Plot:

A density plot can be seen as an extension of the histogram. As opposed to the histogram, the density plot can smooth out the distribution of values and reduce the noise.

```
#Let's take another look at unit price, but this time with a KDE plot.
sns.kdeplot(x = df["unit_price"], fill = True)
plt.show()
```



We can see here that the prices of the pizzas mostly center around $12, 16, and 21. That was worth looking at!

Key Findings:

1.January and April hold the most pizza sales during the year.¶
2. Thursday, Saturday, and Wednesday hold our most pizza sales during the week.
3. 12 PM, 1:30 PM, 5:30 PM and 6:30 PM hold our most pizza sales within the day.
4. The most sold pizza size is Large, followed by Medium and then Small.
5. The most sold pizza type is the classic, and more specifically the Classic Delux Pizza.

CONCLUSION:

The pizza sales analysis reveals key insights into customer preferences and sales trends. Certain pizza types and sizes are consistently popular, indicating demand patterns that can guide menu offerings and promotional strategies. Sales vary by time, with peak sales on weekends and certain months, suggesting opportunities for targeted marketing during high-demand periods. Customer segments also show distinct preferences, allowing for tailored promotions that enhance customer engagement.